

## PERAMALAN BEBAN PEMAKAIAN LISTRIK JAWA TENGAH DAN DAERAH ISTIMEWA YOGYAKARTA DENGAN MENGGUNAKAN *HYBRID AUTOREGRESIVE INTEGRATED MOVING AVERAGE – NEURAL NETWORK*

Berta Elvionita Fitriani<sup>1</sup>, Dwi Ispriyanti<sup>2</sup>, Alan Prahutama<sup>3</sup>

<sup>1</sup>Mahasiswa Jurusan Statistika FSM Universitas Diponegoro

<sup>2,3</sup>Staff Pengajar Jurusan Statistika FSM Universitas Diponegoro

Email: [efberta@yahoo.com](mailto:efberta@yahoo.com) [ispriyanti.dwi@gmail.com](mailto:ispriyanti.dwi@gmail.com) [alan.prahutama@gmail.com](mailto:alan.prahutama@gmail.com)

### ABSTRACT

Excessive use of electronic devices in household and industry has made the demand of nation's electrical power increase significantly these days. As a corporation that aim to provide national electrical power, Perusahaan Listrik Negara (PLN) that distributes electrical power to Central Java and Yogyakarta has to be able to provide an economical and reliable system of electrical power provider. This study aimed to forecast data of electrical power usage in Central Java and Yogyakarta for the next 30 days. There were three forecasting methods used in this study; *Neural Networks* and *Hybrid ARIMA-NN*. The data used in this study was electrical power usage data in January 2014 - November 2014 in Central Java and Yogyakarta. The accuracy of the study was measured based on *MSE* criteria where the best model chosen was the model that has lowest *MSE* value. According to the result of the analysis, using *Neural Networks* model to forecast electrical power usage for the next 30 days has better forecasting result than *Hybrid ARIMA-NN* model.

**Key Word** : *electrical power usage, forecasting of electrical power usage, ARIMA, NN, hybrid ARIMA-NN*

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Kebutuhan akan tenaga listrik saat ini sangatlah tinggi, listrik memegang peranan penting di kehidupan masyarakat saat ini karena peralatan-peralatan elektronik baik dari rumah tangga maupun industri dan perusahaan membutuhkan listrik sebagai tenaga penggerak. Tenaga Listrik tidak dapat disimpan dalam skala besar, sehingga harus disediakan pada saat dibutuhkan. Apabila daya yang dikirim dari pembangkit jauh lebih besar daripada permintaan daya pada beban, maka akan timbul pemborosan energi pada perusahaan listrik. Sedangkan apabila daya yang dibangkitkan dan dikirimkan lebih rendah atau tidak memenuhi kebutuhan konsumen maka akan terjadi pemadaman, yang akibatnya merugikan pihak konsumen. Akibatnya timbul persoalan dalam menghadapi kebutuhan daya listrik yang tidak tetap dari waktu ke waktu dan bagaimana mengoperasikan suatu sistem tenaga listrik yang selalu dapat memenuhi permintaan daya pada setiap saat dengan kualitas baik.

Saat ini, pasokan listrik di Jawa Tengah dan Daerah Istimewa Yogyakarta sangat terbatas bahkan sudah melewati batas minimal pasokan aman. Batas minimal pasokan yang aman adalah 30 persen dari total kebutuhan energi, atau sekitar 12.000 MW dari total kebutuhan 40.000 MW. Sedangkan kondisi cadangan listrik masyarakat Jawa Tengah dan Daerah Istimewa Yogyakarta hanya tinggal 20 persen ([www.dok.joglosemar.co](http://www.dok.joglosemar.co)). Langkah pertama yang harus dilakukan Perusahaan Listrik Negara (PLN) distribusi Jawa Tengah dan Daerah Istimewa Yogyakarta sebagai perusahaan yang dibentuk dengan tujuan untuk menyelenggarakan usaha penyediaan tenaga listrik adalah harus mengetahui beban atau permintaan daya listrik di masa depan dengan melakukan perencanaan operasi sistem yang baik dan tepat. Salah satu langkah perencanaan operasi sistem tenaga listrik yang penting yaitu peramalan kebutuhan beban listrik.

Peramalan adalah kegiatan untuk memperkirakan apa yang akan terjadi di masa yang akan datang. Model *Autoregressive Integrated Moving Average* (ARIMA) dan *Neural Network* (NN) merupakan model yang digunakan dalam peramalan. Model ARIMA merupakan metode peramalan yang bersifat linier. Salah satu syarat data *time series* dimodelkan ARIMA adalah data tersebut harus stasioner. Salah satu pemodelan yang bersifat non linier adalah *Neural Network* (NN). NN merupakan sistem pemrosesan informasi yang mempunyai karakteristik mirip jaringan syaraf biologis. Salah satu algoritma pada NN adalah algoritma *backpropagation*. Algoritma *backpropagation* merupakan algoritma pembelajaran terawasi dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya (*hidden layer*). Untuk mengatasi keterbatasan ARIMA, dapat dibentuk juga model gabungan antara ARIMA dengan model NN yang akan menghasilkan model *hybrid* ARIMA-NN. Menurut Zhang (2003), pada pemodelan *hybrid* ARIMA-NN data dimodelkan dengan ARIMA, sedangkan residual dari model ARIMA dimodelkan dengan NN. Dalam *hybrid* ARIMA-NN terdapat dua komponen yang harus diestimasi dari data, yaitu model ARIMA digunakan untuk menyelesaikan kasus yang linier dan residual dari model yang linier masih mengandung informasi hubungan non-linier dimodelkan menggunakan NN.

Semakin kompleks metode yang digunakan belum tentu metode tersebut menghasilkan akurasi yang lebih baik dibandingkan metode sederhana (klasik). Hal ini disebutkan dalam hasil *M3 Competition* (Makridakis dan Hibon, 2000). Dalam penelitian ini akan digunakan metode ARIMA dan NN sebagai model individu serta model *hybrid* sebagai model kombinasi untuk meramalkan beban pemakaian listrik Jawa Tengah dan Daerah Istimewa Yogyakarta. Keakuratan model dalam peramalan diukur berdasarkan kriteria MSE dimana model terbaik yang dipilih adalah model dengan MSE terkecil.

## 1.2 Tujuan

Tujuan dari penelitian ini adalah mendapatkan model peramalan data Beban Pemakaian Listrik Jawa Tengah dan Daerah Istimewa Yogyakarta dengan menggunakan model *hybrid* ARIMA-NN.

## 2. TINJAUAN PUSTAKA

### 2.1 Analisis Runtun Waktu

*Time series* atau runtun waktu adalah himpunan observasi terurut dalam waktu (Wei, 2006). Analisis runtun waktu atau analisis *time series* dikenalkan pada tahun 1970 oleh George E.P.Box dan Gwilym M. Jenkins melalui bukunya *Time Series Analysis: Forecasting and Control*. Sejak saat itu, *time series* mulai banyak dikembangkan. Dasar pemikiran *time series* adalah pengamatan sekarang ( $Z_t$ ) tergantung pengamatan sebelumnya ( $Z_{t-k}$ ) (Makridakis, 1999). Tujuan analisis *time series* antara lain yaitu memahami dan menjelaskan mekanisme tertentu, meramalkan suatu nilai di masa depan. Analisis *time series* dapat diterapkan di bidang ekonomi, bisnis, industri, teknik dan ilmu-ilmu sosial.

### 2.2 Model ARIMA Box-Jenkins

#### 2.2.1 Model Autoregressive (AR)

Bentuk umum suatu proses *Autoregressive* tingkat  $p$  (AR( $p$ )) dalam Soejoeti (1987) adalah :

$$Z_t = \phi_1 Z_{t-1} + \phi_2 Z_{t-2} + \dots + \phi_p Z_{t-p} + a_t \quad (1)$$

Definisi Persamaan (1) adalah nilai sekarang suatu proses dinyatakan sebagai jumlah tertimbang nilai-nilai yang lalu ditambah satu sesatan sekarang.

### 2.2.2 Model *Moving Average* (MA)

Bentuk umum model *Moving Average* tingkat q atau MA(q) adalah :

$$Z_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \dots - \theta_q a_{t-q} \quad (2)$$

### 2.2.3 Model ARIMA (p,d,q)

Model umum *autoregressive* orde p, *Integrated* orde d, dan *Moving Average* orde q (ARIMA(p,d,q)) merupakan hasil penggabungan dengan proses non stasioner yang telah distasionerkan. Dalam hal ini, d merupakan orde dari *differencing*. Persamaan umum:

$$\phi_p(B)(1-B)^d Z_t = \theta_q(B) + a_t \quad (3)$$

dimana

$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$  merupakan operator AR(p)

$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$  merupakan operator MA(q)

## 2.3 Runtun Waktu Musiman

Secara umum bentuk model *Seasonal Autoregressive Integrated Moving Average* (SARIMA (p,d,q)(P,D,Q)<sup>s</sup>)

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D Z_t = \theta_q(B)\Theta_Q(B^s)a_t \quad (4)$$

dimana:

p,d,q = tingkat AR, *differencing* non-musiman, dan MA

P,D,Q = tingkat AR, *differencing* musiman, dan MA

$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$

$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \dots - \Phi_P B^{Ps}$

$(1-B)^d$  = tingkat *differencing* non-musiman

$(1-B^s)^D$  = tingkat *differencing* musiman

$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$

$\Theta_Q(B^s) = 1 - \theta_1 B^s - \theta_2 B^{2s} - \dots - \theta_Q B^{Qs}$

## 2.4 Neural Network

Warsito (2009) menjelaskan Jaringan Syaraf Tiruan (JST) atau yang dikenal dengan istilah *Neural Network* (NN) adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologis.

Menurut Kusumadewi (2003) berdasarkan karakteristik jaringan syaraf manusia, dibangun suatu mesin pembelajaran NN yang memiliki komponen-komponen yang hampir sama dengan komponen pada jaringan syaraf manusia, diantaranya:

### 1) Neuron/Node

Komponen NN yang bertugas memproses informasi, memiliki fungsi yang sama dengan neuron pada otak manusia, dimana semua proses perhitungan dilakukan disini (Kusumadewi, 2003).

### 2) Input

Pandjaitan (2007) menerangkan bahwa informasi yang akan diproses oleh neuron *input* ini bisa berasal dari lingkungan ataupun dari *node* lain.

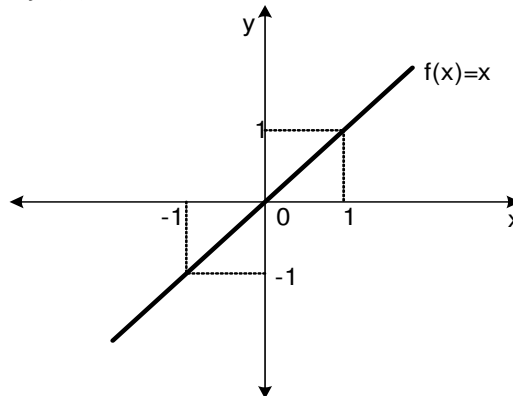
### 3) Fungsi Aktivasi

Dalam NN fungsi aktivasi akan menentukan *output* suatu unit (mengubah sinyal *input* menjadi sinyal *output*) yang akan dikirim ke unit lainnya. Beberapa fungsi aktivasi yang umumnya dipakai dalam NN adalah sebagai berikut:

#### a. Fungsi Identitas (Linier/*Purelin*)

Menurut Siang (2009) fungsi identitas merupakan suatu fungsi yang memiliki nilai *output* sama dengan nilai *input*. Fungsi identitas dirumuskan sebagai berikut:

$$f(x) = x, -\infty < x < \infty$$

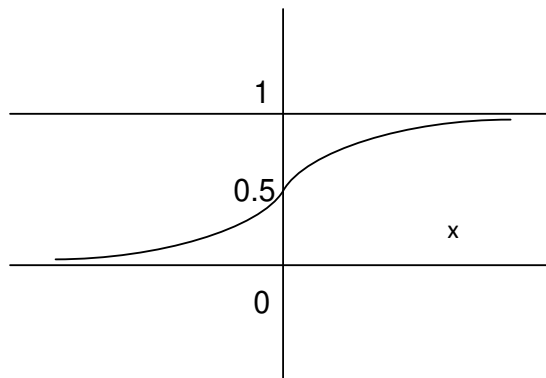


**Gambar 1.** Fungsi identitas (linier)

#### b. Fungsi Sigmoid Biner (Sigmoid Logistik)

Menurut Kusumadewi (2003) fungsi sigmoid biner ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner dirumuskan sebagai berikut (Siang, 2009):

$$f(x) = \frac{1}{1+e^{-x}} = [0,1], -\infty < x < \infty$$



**Gambar 2.** Fungsi Sigmoid Biner (sigmoid logistik)

### 4) Bobot

Nilai yang menunjukkan kekuatan hubungan antar *node*. Jika nilai bobot diatur sedemikian rupa, maka dihasilkan *output* yang berbeda juga. Bobot-bobot yang optimal akan memungkinkan sistem menterjemahkan data masukan secara benar dan menghasilkan keluaran yang diinginkan (Siang, 2009). Salah satu pembobotan dalam NN yaitu dengan cara inisialisasi bobot. Inisialisasi bobot digunakan untuk menentukan nilai-nilai bobot pada penghubung antar neuron. Setiap membentuk jaringan *backpropagation*, Matlab akan memberi nilai bobot dan bias awal dengan bilangan acak. Bobot dan bias ini akan berubah setiap kali membentuk jaringan.

## 5) Output

Suatu nilai yang merupakan hasil dari fungsi aktivasi, yang bisa berupa *output* dari jaringan atau menjadi *input* bagi *node* lain. Data *output* merupakan data numerik.

Arsitektur jaringan dalam NN merupakan susunan dari neuron-neuron dalam lapisan dan pola keterhubungannya dalam dan antar lapisan. Sebuah NN biasanya terdiri atas lapisan-lapisan antara lain:

- Lapisan masukan (*Input layer*)  
yaitu lapisan yang menerima masukan/*input* dari jaringan luar.
- Lapisan tersembunyi (*Hidden layer*)  
yaitu lapisan yang terletak dalam satu atau beberapa lapisan dan tidak berhubungan langsung dengan keadaan di luar jaringan. Lapisan tersembunyi biasanya terletak di antara lapisan *input* dan lapisan *output*. Menurut Fahmi (2011) dalam Yasin (2014) berikut tabel rumus penentuan banyaknya neuron pada *hidden layer* dengan  $N_{in}$  adalah neuron *Input*:

**Tabel 1. Penentuan Banyaknya Neuron Hidden Layer**

$N_{out}$ (Neuron output)	Rumus menghitung <i>hidden layer</i>
1	$2 * N_{in} + 1$
2	$3 * N_{in}$
3	$\frac{2 + N_{in} * N_{out} + 0.5 N_{in} * (N_{out}^2 + N_{in}) - 3}{N_{in} + N_{out}}$
4	$\frac{(2 * N_{in})}{3}$
5	$\sqrt{N_{in} * N_{out}}$
6	$2 * N_{in}$

- Lapisan keluaran (*Output layer*)  
yaitu lapisan yang menghasilkan *output* dari jaringan.

## 2.5 Backpropagation

*Backpropagation* merupakan algoritma pembelajaran yang terawasi biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya (Kusumadewi, 2003). *Backpropagation* memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi. Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi adalah sebagai berikut:

**Langkah 0:** Inisialisasi bobot

**Langkah 1:** Tetapkan maksimum iterasi, target *error*, dan laju percepatan ( $\alpha$ ).

Fase I : Propagasi Maju

**Langkah 2:** Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.

**Langkah 3:** Menghitung semua keluaran di unit tersembunyi  $z_h$  ( $h = 1, 2, \dots, H$ )

$$z_{net_h} = v_{ho} + \sum_{g=1}^G x_g v_{hg}$$

$$z_h = f(z_{net_h}) = \frac{1}{1 + e^{-z_{net_h}}}$$

**Langkah 4:** Menghitung semua keluaran jaringan di unit  $Y_u$  ( $u = 1, 2, \dots, U$ )

$$y_{net_u} = w_{uo} + \sum_{h=1}^H z_h w_{uh}$$

$$Y_u = f(y_{net_u}) = \frac{1}{1 + e^{-y_{net_u}}}$$

Fase II : Propagasi mundur

**Langkah 5:** Menghitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran  $Y_u$  ( $u = 1, 2, \dots, U$ )

$$\delta_u = (t_u - Y_u) f'(y_{net_u}) = (t_u - y_u) y_u (1 - y_u)$$

$\delta_u$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 6).

Menghitung suku perubahan bobot  $w_{uh}$  yang akan dipakai nanti untuk merubah bobot  $w_{uh}$  dengan laju percepatan  $\alpha$

$$\Delta w_{uh} = \alpha \delta_u z_h ; u = 1, 2, \dots, U ; h = 0, 1, \dots, H$$

**Langkah 6:** Menghitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi  $z_h$  ( $h = 1, 2, \dots, H$ )

$$\delta_{net_h} = \sum_{u=1}^U \delta_u w_{uh}$$

Faktor  $\delta$  unit tersembunyi :

$$\delta_h = \delta_{net_h} f'(z_{net_h}) = \delta_{net_h} z_h (1 - z_h)$$

Menghitung suku perubahan bobot  $v_{hg}$  yang akan dipakai nanti untuk merubah bobot  $v_{hg}$ .

$$\Delta v_{hg} = \alpha \delta_h x_g ; h = 1, 2, \dots, H ; g = 1, 2, \dots, G$$

Fase III : Perubahan bobot

**Langkah 7:** Menghitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran :

$$w_{uh}(\text{baru}) = w_{uh}(\text{lama}) + \Delta w_{uh} ; u = 1, 2, \dots, U ; h = 0, 1, \dots, H$$

Perubahan bobot garis yang menuju ke unit tersembunyi :

$$v_{hg}(\text{baru}) = v_{hg}(\text{lama}) + \Delta v_{hg} ; h = 1, 2, \dots, H ; g = 0, 1, \dots, G$$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

## 2.6 Hybrid ARIMA-NN

*Hibryd* adalah kombinasi dua atau lebih sistem dalam satu fungsi, dalam hal ini adalah kombinasi antara ARIMA dengan *Neural Network*. Banyak peneliti menggunakan metode *hybrid* karena diharapkan dapat saling melengkapi karena dalam dunia nyata jarang ditemukan kejadian *time series* yang murni linier ataupun murni non-linier (Zheng & Zhong, 2011). Secara umum kombinasi dari model *time series* yang linier dan non-linier dapat dituliskan sebagai berikut :

$$y_t = L_t + E_t \quad (5)$$

dimana  $L_t$  menunjukkan komponen linier dan  $E_t$  menunjukkan komponen non linier. Terdapat dua komponen yang harus diestimasi dari data, yaitu misalkan model ARIMA digunakan untuk menyelesaikan kasus yang linier, dimana residual dari model yang linier masih mengandung informasi hubungan non-linier. Dapat dituliskan :

$$e_t = y_t - \hat{L}_t \quad (6)$$

dimana  $\hat{L}_t$  nilai ramalan ARIMA pada waktu t. Persamaan dari residual untuk model NN dapat dituliskan sebagai berikut:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (7)$$

dimana f adalah fungsi non linier yang dijelaskan oleh NN dan  $\varepsilon_t$  adalah *error* yang acak, sehingga fungsi kombinasi untuk memprediksi adalah:  $\hat{y}_t = \hat{L}_t + \hat{E}_t$ .

## 2.7 Pemilihan Model Terbaik

Pemilihan model terbaik dalam peramalan ditentukan oleh error yang dihasilkan. Dalam penelitian ini, digunakan MSE (*Mean Square Error*) sebagai kriteria penentuan model terbaik dari beberapa model yang terpilih. Dalam Makridakis (1999) perumusan kriteria MSE dinyatakan sebagai berikut:

$$MSE = \frac{\sum_{t=1}^n e_t^2}{n} \quad (8)$$

dengan

$$e_t = (Z_t - \hat{Z}_t)$$

dimana:

$n$  = banyaknya data

$e_t$  = residual pada periode t

$Z_t$  = data aktual pada periode t

$\hat{Z}_t$  = nilai hasil peramalan pada periode t.

## 3. METODOLOGI PENELITIAN

### 3.1 Data

Data dalam penelitian ini adalah data sekunder yang diperoleh dari PT PLN (Persero) Distribusi Jawa Tengah dan Daerah Istimewa Yogyakarta. Data yang digunakan yaitu data beban puncak pemakaian listrik Jawa Tengah dan Daerah Istimewa Yogyakarta bulan Januari 2014 hingga November 2014.

### 3.2 Metode Analisis

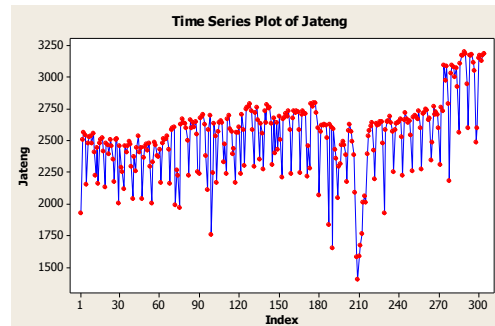
Langkah-langkah dalam peramalan beban penggunaan listrik di Jateng-DIY adalah sebagai berikut:

- 1) Pengumpulan data beban penggunaan listrik Jawa Tengah dan Daerah Istimewa Yogyakarta pada pukul 10.00 WIB bulan Januari 2014 hingga November 2014
- 2) Membagi data menjadi dua bagian yaitu data *in sample* pada Januari 2014 hingga Oktober 2014 dan data *out sample* pada November 2014. Data *in sample* digunakan untuk pembentukan model (*training*), sedangkan data *out sample* digunakan untuk pemilihan ramalan terbaik (*testing*).
- 3) Mengaplikasikan model ARIMA dan *Hybrid* ARIMA-NN pada data penggunaan beban listrik.
- 4) Mendapatkan hasil peramalan beban penggunaan listrik dengan model *Hybrid* ARIMA-NN.

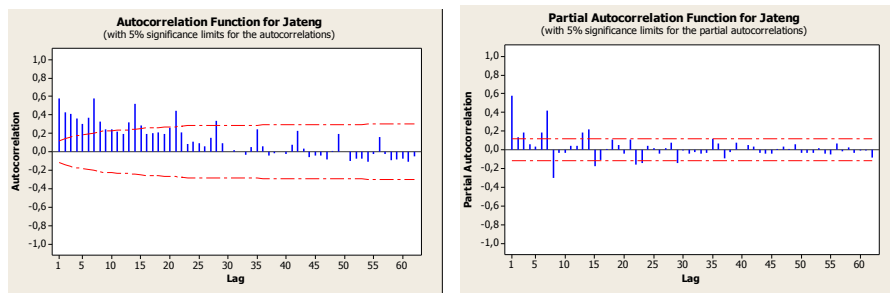


## 4. HASIL DAN PEMBAHASAN

### 4.1 Model Peramalan dengan ARIMA

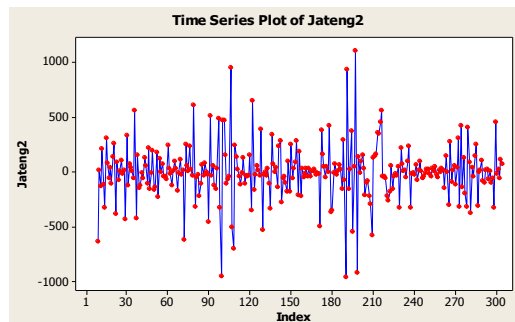


Gambar 3. Plot *Time Series* Beban Pemakaian Listrik

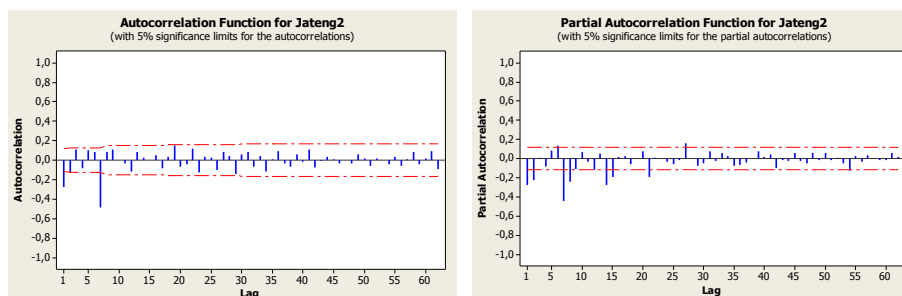


Gambar 4. Plot ACF dan PACF Data Awal ( $Z_t$ )

Berdasarkan Gambar 3, terlihat bahwa data beban pemakaian listrik Jateng-DIY Januari 2014 hingga Oktober 2014 belum stasioner. Gambar 3 menunjukkan bahwa data tidak stasioner dalam *mean* karena berpola *trend* naik an Gambar 4 menunjukkan pola ACF yang turun secara lambat, sehingga perlu dilakukan penanganan berupa *differencing* orde ke-1 dan ke-7 karena data membentuk pola musiman setiap 7 hari.



Gambar 5. Plot *Time Series* setelah *Differencing*



Gambar 6. Plot ACF dan PACF setelah *Differencing*



Diperoleh model terbaik SARIMA (2,1,0)(0,1,1)<sup>7</sup> dari data beban pemakaian listrik Jateng-DIY dengan persamaan model sebagai berikut:

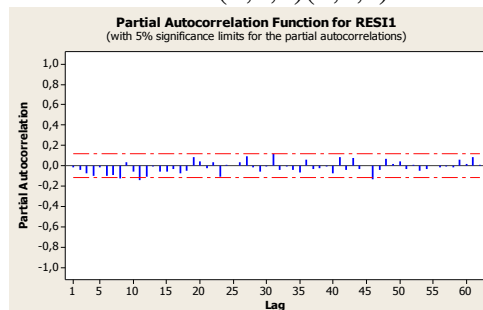
$$Z_t = 0,5805Z_{t-1} + 0,2121Z_{t-2} + 0,2074Z_{t-3} + Z_{t-7} - 0,5805Z_{t-8} - 0,2121Z_{t-9} - 0,2074Z_{t-10} + a_t - 0,9585a_{t-7}$$

#### 4.2 Model Peramalan dengan *Hybrid* ARIMA-NN

Berdasarkan analisis ARIMA diperoleh model terbaik yang digunakan dalam peramalan beban penggunaan listrik, yaitu model SARIMA (2,1,0)(0,1,1)<sup>7</sup> dapat dituliskan sebagai berikut :

$$Z_t = 0,5805Z_{t-1} + 0,2121Z_{t-2} + 0,2074Z_{t-3} + Z_{t-7} - 0,5805Z_{t-8} - 0,2121Z_{t-9} - 0,2074Z_{t-10} + a_t - 0,9585a_{t-7}$$

Kemudian dilihat lag yang *cut off* pada plot PACF dari residual yang diperoleh dari model SARIMA (2,1,0)(0,1,1)<sup>7</sup>. Berdasarkan plot PACF terlihat pada Gambar 7 bahwa lag *cut off* pada lag ke-8. Oleh karena itu digunakan lag 8 sebagai *input* pada permodelan NN untuk meramalkan residual dari model SARIMA (2,1,0)(0,1,1)<sup>7</sup>.



**Gambar 7.** Plot PACF Residual SARIMA (2,1,0)(0,1,1)<sup>7</sup>

Peramalan residual model SARIMA (2,1,0)(0,1,1)<sup>7</sup> dengan menggunakan NN dilakukan dengan *input* lag 8 yang berarti terdapat 1 buah neuron pada unit masukan dan jumlah unit pada *hidden layer* sebanyak 3 neuron berdasarkan hasil perhitungan pada Tabel 1. Sehingga diperoleh hasil peramalan dengan menggunakan *hybrid* ARIMA-NN untuk data beban pemakaian listrik Jateng-DIY pada bulan November 2014 dengan MSE sebesar 188330,694.

#### 5. KESIMPULAN

- 1) Berdasarkan pemodelan menggunakan ARIMA diperoleh model terbaik dengan parameter-parameter yang signifikan terhadap model, memenuhi asumsi independensi residual, dan dengan MSE sebesar 26492 yaitu SARIMA (2,1,0)(0,1,1)<sup>7</sup>.
- 2) *Hybrid* ARIMA-NN yang merupakan gabungan dari model ARIMA dengan *neural network* tidak menjamin kinerja hasil peramalan yang lebih baik. Seperti yang disebutkan dalam hasil *M3 Competition*, semakin kompleks metode yang digunakan belum tentu metode tersebut menghasilkan akurasi yang lebih baik dibandingkan metode sederhana (klasik).

#### DAFTAR PUSTAKA

- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- Makridakis, S., & Hibon, M. 2000. *The M3-Competition: Results, Conclusions and Implications*. International Journal of Forecasting. 16: Hal.451-476.

- Makridakis, S., Wheelwright, S. C., McGee, V. E. 1999. *Metode dan Aplikasi Peramalan*. Edisi Kedua. Untung Sus Adriyanto dan Abdul Basith, penerjemah. Jakarta: Erlangga. Terjemahan dari: *Forecasting, 2nd Edition*.
- Pandjaitan, L.W. 2007. *Dasar-Dasar Komputasi Cerdas*. Yogyakarta: Andi Offset.
- Siang, J.J. 2009. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta: Andi Offset.
- Soejoeti, Z. 1987. *Analisis Runtun Waktu*. Jakarta: Karunika.
- Warsito, B. 2009. *Kapita Selekta Statistika Neural Network*. Semarang: BP Universitas Diponegoro.
- Yasin, H., Suparti. 2014. *Pemodelan Volatilitas untuk Penghitungan Value at Risk Menggunakan Feed Forward Neural Network dan Algoritma Genetika*. Media Statistik Vol.7, No.2: Hal.53-61.
- Zhang, G.P. 2003. Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 50: Hal.159-175.
- Zheng, F. & Zhong, S. 2011. Time Series Forecasting Using a Hybrid ARIMA RBF Neural Network Model and AR Model Based on Binomial Smoothing. *World Academy of Science and Technology*, Hal.75.
- [www.dok.joglosemar.co](http://www.dok.joglosemar.co) (diakses pada tanggal 27 Mei 2015)